Simulation of Benthic Ripples and Transport Processes for SAX

Oliver B. Fringer
Terman Bldg., Rm M17
Dept. of Civil and Environmental Engineering
Stanford University, Stanford, CA 94305-4020
phone: (650) 725-6878 fax: (650) 725-9720 email: fringer@stanford.edu

Robert L. Street
Terman Bldg., Rm M13
Dept. of Civil and Environmental Engineering
Stanford University, Stanford, CA 94305-4020
phone: (650) 723-4969 fax: (650) 725-9720 email: street@stanford.edu

Award Number: N00014-05-1-0177

LONG-TERM GOALS

Our goal is to provide a complete simulation code that will represent and predict the sediment transport and bed features on the continental shelf at user-specified resolution by using state-of-the-art algorithms for the physics and numerics of the simulation code.

OBJECTIVES

Our primary objective is to simulate the ripple climate on the bed of the inner shelf at depths on the order of 20 m and over domains ranging from centimeters to kilometers in support of the SAX experiments and analyses. Our secondary objective is to use simulation to better understand the physics of ripple formation and sediment transport in this environment. A third objective is to create a powerful and fast parallel numerical code system by the synthesis of a number of our computational tools.

APPROACH

The Critical Benthic Environmental Processes and Modeling at SAX04 (aka Ripples) Departmental Research Initiative of ONR has a number of objectives. Among them our work is addressing the following:

- Model ripple morphology and its gradients on scales ranging up to kilometers. Model the effects of bioturbation on the ripple evolution.

- Understand the response of ripples to changes in wave and wave-current forcing.

Based on our experience in oceanic and atmospheric field-experiment programs, we also understand the importance of the close and timely collaboration of all of the investigators in the modeling and field components of the program. For the DRI we are aiming to be able to make predictive simulations of the bed evolution for the second field season in FY06. We are using the conditions from SAX04 as the basis for hindcasts and code test and verification using the synthesis of codes described below.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **30 SEP 2005**	2. REPORT TYPE	3. DATES COVERED **00-00-2005 to 00-00-2005**

4. TITLE AND SUBTITLE **Simulation of Benthic Ripples and Transport Processes for SAX**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Stanford University,Stanford,CA, 94305**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
code 1 only

14. ABSTRACT
Our primary objective is to simulate the ripple climate on the bed of the inner shelf at depths on the order of 20 m and over domains ranging from centimeters to kilometers in support of the SAX experiments and analyses. Our secondary objective is to use simulation to better understand the physics of ripple formation and sediment transport in this environment. A third objective is to create a powerful and fast parallel numerical code system by the synthesis of a number of our computational tools.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**8**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Field-scale solver We are using the parallel unstructured grid code, known as SUNTANS (Stanford Unstructured Nonhydrostatic Terrain-following Adaptive Navier-Stokes Simulator), which solves the nonhydrostatic Navier-Stokes equations under the Boussinesq approximation with the large-eddy formulation for the resolved motions. The code has been developed by Fringer *et al.* (2005) in order to determine the behavior of the nonhydrostatic internal wave spectrum in Monterey Bay (Fringer *et al.*, 2004). SUNTANS is a high-resolution simulation tool for coastal, estuarine, and river simulations, and incorporates a host of processes, from the free-surface to the bed, that play an important role in driving smaller-scale motions, including the nonhydrostatic pressure, wind stress, surface and internal tides, and parameterizations of small-scale internal as well as surface breaking physics. Most importantly, because it is unstructured, and adaptive in the long-term, we have the ability to employ higher resolution around the site of interest for the SAX field experiments where we embed the smaller-scale simulation codes that we describe below.

Parallel large-eddy simulation code The SUNTANS code is being used to determine the field-scale motions that drive the smaller-scale flow, which is computed with a nonhydrostatic parallel large-eddy simulation Navier-Stokes solver. This solver employs the code of Cui and Street (2001; 2004) (henceforth referred to as PCUI) which is an adaptation of the large-eddy simulation solver of Zang *et al.* (1994) to simulate laboratory-scale rotating stratified flows using MPI, the message passing interface, on parallel computers. The formulation of Zang *et al.* employs a fully non-orthogonal curvilinear, boundary-following grid as well as a dynamic-mixed subfilter scale model (Zang *et al.*, 1993) to compute the subfilter scale terms that arise from volume filtering. The equations are discretized in time with a semi-implicit, second-order accurate method and in space with second-order differencing and advanced with a fractional step/projection method. The method has been applied to a host of simulations, including turbulent stratified flow over a wavy bed (Calhoun *et al.*, 1999), interfacial breaking waves (Fringer & Street, 2003), and the study of a round jet in crossflow (Yuan *et al.*, 1999) using a total of 20 overlapping grids. Along with having been verified extensively, the PCUI code is highly efficient, as it was 41% faster than the next comparable code on the NASA IBM SP-2 (Bergeron, 1998).

Sediment transport For sediment transport, we are employing algorithms from the code of Zedler and Street (2001; 2002; 2005), who extended the formulation of Zang *et al.* (1994) and added a module which solves the advection diffusion equation (with settling term) for the sediment. Under this work Zedler and Street also created a bed-load module and a bed erosion/deposition module (Zedler, personal communication, 2005) to go with the module for suspended sediment transport. She provided us with the equations in the useful terrain-following coordinates of her original code so that the resolution of shear stress direction, for example, is clear.

Motion of the bed may be computed using the immersed boundary method (IBM), which allows the tracking of complex boundaries on Cartesian grids (Tseng & Ferziger, 2003), or by the moving boundary method of Hodges and Street (1999) and Fringer (2003) which preserves the near-bed terrain following coordinates. The IBM method effectively imposes forcing functions on the right hand side of the momentum equations that enforce boundary conditions at complex water-ripple boundaries; under the related project cited below, Zedler has created a preliminary implementation of an IBM for flow and sediment transport at a bed.

The Zedler code was a serial code and large problems ran very slowly; thus, the elements of the Zedler and Street code are currently being implemented into the PCUI code to run in parallel mode. This code is referred to as PCUI_Sed. Both options for moving the boundary are being kept open at this time.

Strategy: We are molding the above tools into an integrated system wherein:

- An enhanced SUNTANS drives the sediment transport code PCUI_Sed to simulate local ripple climate wherever that information is needed. This need not be done in real time, but can be done rapidly because the codes are parallelized.
- The parallelized, IBM sediment transport code (PCUI_Sed) simulates the local bed evolution. Our plan is to run the nested PCUI_Sed in a domain of roughly 10 m by 10 m with 5 cm horizontal resolution. Results from this domain could in principle drive a 50 cm by 50 cm domain with 25 mm horizontal resolution. However, based on SAX99 information, we anticipate a focus on ripple lengths ranging from about 0.5 m down to 7 cm.

WORK COMPLETED

This grant began in January 2005. At this time we are in the process of implementing the immersed boundary method into SUNTANS to create a terrain-following boundary. We report on this implementation below. Our next steps involve modification of the PCUI parallel code to create PCUI_Sed using the tools created by Zedler under the related project cited below.

RESULTS

The immersed boundary method employs the use of Cartesian grids in complex geometries by adjusting ghost cell velocity and pressure values in such a way that boundary conditions are satisfied on the real, or immersed, physical boundaries, rather than at the faces of the Cartesian grid. The ghost-cell velocities and pressure field are extrapolated from the computational cells inside the immersed boundary while satisfying boundary conditions at that boundary. On staggered grids, which are employed in SUNTANS, the extrapolation procedure depends on whether the immersed boundary lies between vertical or horizontal faces, and it is quite different for the two. For horizontal faces, the computation in the absence of the immersed boundary assumes that the bottom boundary coincides with the face of a Cartesian grid cell, as depicted in Figure 1(a). Inclusion of the immersed boundary effectively alters the position of the no-slip boundary condition and as a result the value of the ghost-cell velocity is altered to account for this, as depicted in Figure 1(b). For vertical faces, the horizontal velocity vectors at the vertical faces of the Cartesian boundary are zero in the absence of the immersed boundary, while its inclusion adds nonzero velocities to these faces. This in turn alters the horizontal and vertical velocity field to account for the complex geometry more accurately.

The actual implementation of the immersed boundary method is referred to in the literature as ghost velocity reconstruction (Tseng & Ferziger, 2003). Using the procedure of Tseng and Ferziger, the ghost values beneath the bottom boundary are computed through linear extrapolation, which is advantageous on the unstructured, z-level grids in SUNTANS (see Figure 4 for a schemetic of the z-level prism cells) because it only requires velocities on the upper and lower sides of the immersed boundary rather than at other faces of the unstructured grid, which would require complicated and impractical interpolation procedures. Using linear extrapolation in the vertical has the added advantage of guaranteeing that the ghost velocity is normal to the vertical face where it is located.

Following the work of Kim *et al.* (2001), the construction method at bottom boundaries is illustrated in Figure 3. Here, $\boldsymbol{U}$ represents the component of the velocity normal to vertical faces, while $\boldsymbol{d_0}$, $\boldsymbol{d_1}$

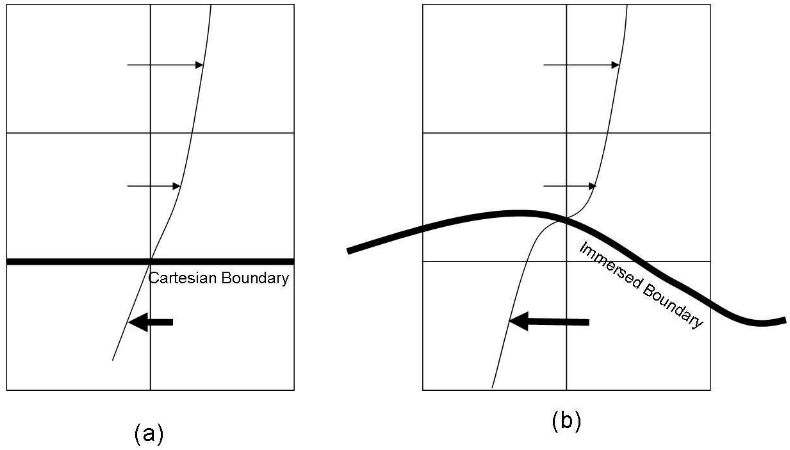

Figure 1: Depiction of the immersed boundary method as applied to horizontal faces showing its effect on the ghost velocities (in bold) without (a) and with (b) the immersed boundary.

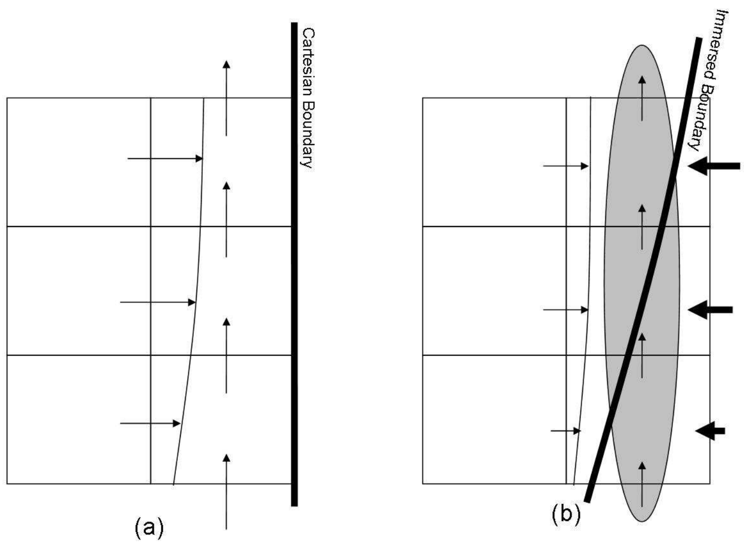

Figure 2: Depiction of the immersed boundary method as applied to vertical faces showing its effect on the ghost velocities (in bold) without (a) and with (b) the immersed boundary. The shaded region shows that the immersed boundary also indirectly affects the vertical velocity field in the cells containing the immersed boundary.

and d_2 represent the distance between the immersed boundary (IB) and the ghost point, the center of the bottommost cell inside fluid domain, and the center of the cell right above the bottommost cell, respectively. In order to ensure stability, the ghost velocity must be bounded by the minimum and maximum values of the velocity field within the immersed boundary. This is ensured if the interpolation is altered when the distance between the ghost velocity and the immersed boundary exceeds d_1, such that, if U_g is the ghost velocity and U_k and U_{k-1} are the velocities above the immersed boundary, then

$$U_g = \begin{cases} -\frac{d_0}{d_1} U_k & 0 < d_0 \le d_1\,, \\ -\frac{(d_2-d_0)U_k+(d_0-d_1)U_k-1}{d_2-d_1} & \text{otherwise}\,. \end{cases}$$

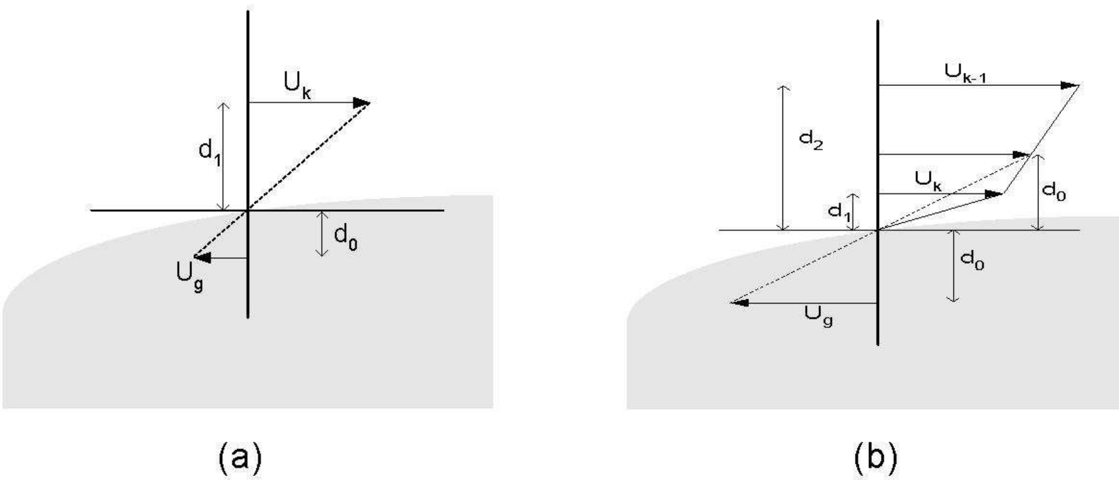

Figure 3: Depiction of the immersed boundary method of obtaining the ghost velocity U_g for the case when (a) the ghost velocity is closer to the immersed boundary than the computational velocity and (b) the ghost velocity is farther away from the immersed boundary than the computational velocity.

Because SUNTANS grids are unstructured in plan, the formulation is much more complex for the vertical faces, since boundary conditions must be satisfied at piecewise planar immersed boundaries that intersect prism cells, as depicted in Figure 4. The method is similar to that depicted in Figure 3 except the interpolation is much more complex because of the unstructured nature of the horizontal interpolation.

While the velocity field must satisfy Dirichlet conditions at the immersed boundaries, the pressure field is updated by satisfying a Neumann condition, since there must be a zero pressure gradient normal to immersed boundaries. This is naturally satisfied for the pressure field because it is specified at the centers of the cells rather than at the faces, which complicates the immersed boundary method for the velocity field. To compute the value of the pressure field in the ghost cells, we assume the immersed boundary that intersects an unstructured grid cell is given by the piecewise-planar function (given by the red lines in Figure 4)

$$ax + by + cz = d\,, \tag{1}$$

such that $\sqrt{a^2+b^2+c^2} = 1$. Following Tseng and Ferziger (2003), if the value of the pressure in the vicinity of the boundary is assumed to be of the form

$$p(x, y, z) = w_1 x + w_2 y + w_3 z + w_4\,, \tag{2}$$

then it follows that the value of the pressure at the ghost point in Figure 5 is

$$p_g = w_1 x_g + w_2 y_g + w_3 z_g + w_4\,, \tag{3}$$

while the pressure gradient is $\nabla P = w_1 \mathbf{e}_x + w_2 \mathbf{e}_y + w_3 \mathbf{e}_z$, where $\mathbf{e}_{x,y,z}$ are the Cartesian unit vectors. If $\mathbf{n}_b$ represents the normal to the plane defined in equation (1), then the component of the pressure gradient in the direction of $\mathbf{n}_b$ is given by

$$\left.\frac{\partial p}{\partial n}\right|_b = \nabla P \cdot \mathbf{n}_b = a w_1 + b w_2 + c w_3 \,.$$

Using this equation, along with three constraints that require the pressure at points $\mathbf{1}$, $\mathbf{2}$, and $\mathbf{3}$ in Figure 5 to satisfy equation (2), one arrives at the governing equations for the weights in equation (3) as

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \begin{bmatrix} a & b & c & 0 \\ x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \left.\frac{\partial p}{\partial n}\right|_b \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} .$$

Here, p_1, p_2, and p_3 are the pressure in the computational cells inside the immersed boundary which are located at $(x, y, z)_1$, $(x, y, z)_2$, and $(x, y, z)_3$, respectively, as in Figure 5, and $(\partial p/\partial n)_b = 0$ is the desired pressure gradient normal to the immersed boundary.

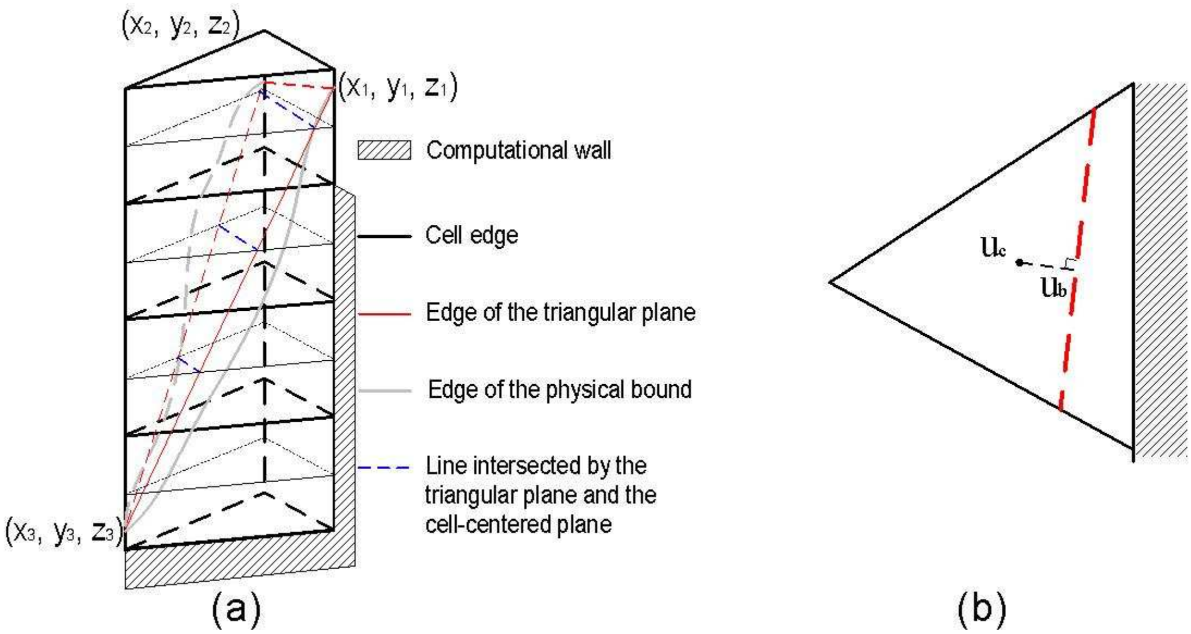

Figure 4: Depiction of how the bottom topography (in gray) intersects the triangular prisms in piecewise-planar faces (red). (a) Three-dimensional view, (b) top view, showing the velocity vector defined at the cell center $\mathbf{u}_e$, and the velocity interpolated to the immersed boundary edge $\mathbf{u}_b$.

RELATED PROJECTS

This project is linked to the just completed ONR Grant N00014-00-1-0440 “Large Eddy Simulation of Sediment Transport in the Presence of Surface Gravity Waves, Currents and Complex Bedforms” from which we are adapting algorithms created by Dr. Zedler for sediment transport and bed movement.

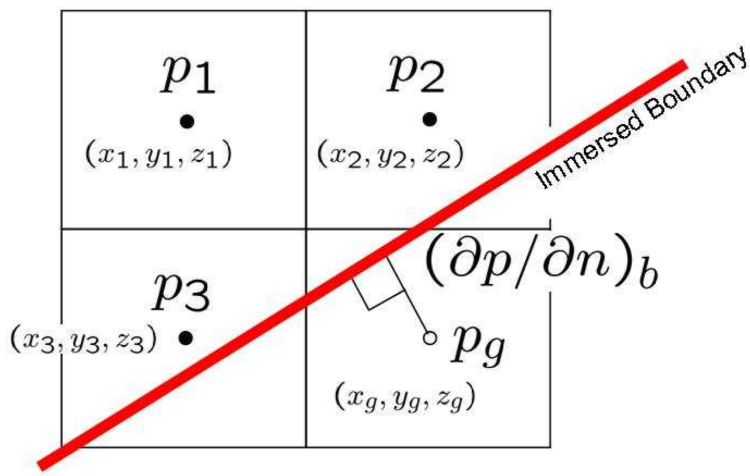

Figure 5: Vertical slice depicting how the normal pressure gradient condition is satisfied at the immersed boundary using the pressure at three computational cells. The red line represents the intersection of the red plane in Figure 4 with the vertical plane represented here.

HONORS/AWARDS/PRIZES

R. L. Street: 2005 Hunter Rouse Hydraulic Engineering Award, ASCE
FY 2005 Japan Society for the Promotion of Science Fellowship for Research in Japan

REFERENCES

Bergeron, R. 1998. Measurement of a scientific workload using the IBM hardware performance monitor. *Proceedings of the Supercomputing'98 Conference*, *Orlando*.

Calhoun, R. J., Street, R. L., & Koseff, J. R. 1999. Turbulent flow over a wavy surface: Stratified case. *J. Geophys. Res.*, **106**, 9295–9310.

Cui, A., & Street, R. L. 2001. Large-eddy simulation of turbulent rotation convective flow development. *J. Fluid Mech.*, **447**, 53–84.

Cui, A., & Street, R. L. 2004. Large-eddy simulation of coastal upwelling flow. *Environmental Fluid Mechanics*, **4**, 197–223.

Fringer, O. B. 2003. *Numerical simulations of breaking interfacial waves*. Ph.D. Dissertation, Stanford University.

Fringer, O. B., & Street, R. L. 2003. The dynamics of breaking progressive interfacial waves. *J. Fluid Mech.*, **494**, 319–353.

Fringer, O. B., Gerritsen, M., & Street, R. L. 2004. Internal waves in Monterey Bay: an application of SUNTANS. *Proceedings of the 4th International Symposium on Environmental Hydraulics*, Hong Kong.

Fringer, O. B., Gerritsen, M., & Street, R. L. 2005. An unstructured-grid, finite-volume, nonhydrostatic, parallel coastal ocean simulator. *Ocean Modelling*. submitted.

Hodges, B. R., & Street, R. L. 1999. On simulation of turbulent nonlinear free-surface flows. *J. Comput. Phys.*, **151**, 425–457.

Kim, J., Kim, D., & Choi, H. 2001. An immersed-boundary finite-volume method for simulations of flows in complex geometries. *J. Comp. Phys.*, **171**, 132–150.

Tseng, Y. H., & Ferziger, J. H. 2003. A ghost-cell immersed boundary method for flow in complex geometry. *J. Comput. Phys.*, **192**, 593–623.

Yuan, L., Street, R. L., & Ferziger, J. H. 1999. Large-eddy simulations of a round jet in crossflow. *J. Fluid Mech.*, **379**, 71–104.

Zang, Y., Street, R. L., & Koseff, J. R. 1993. A dynamic mixed subgrid-scale model and its application to turbulent recirculating flows. *Phys. Fluids A*, **5**, 3186–3196.

Zang, Y., Street, R. L., & Koseff, J. R. 1994. A non-staggered grid, fractional step method for time-dependent incompressible Navier-Stokes equations in curvilinear coordinates. *J. Comput. Phys.*, **114**, 18–33.

Zedler, E. A., & Street, R. L. 2001. Large-eddy simulation of sediment transport: currents over ripples. *J. Hydraul. Eng.*, **127**, 442–452.

Zedler, E. A., & Street, R. L. 2002. Nearshore sediment transport processes: unearthed by Large Eddy Simulation. *Pages 2504–2516 of: 28th International Conference on Coastal Engineering, ASCE*.

Zedler, E. A., & Street, R. L. 2005. Sediment transport over vortex ripples in oscillatory flow. *J. Hydraul. Eng.* in press.